WHAT IS CLAIMED IS:

1. A method of processing the driving program of a smart peripheral device in a computer system using an operating system, wherein the smart peripheral device is connected to the computer system through a peripheral bus and the operating system includes a bus-driving program for controlling the peripheral bus, the processing method comprising:

requesting that the operating system return any data regarding a connectivity status of a particular peripheral device connected to the peripheral bus;

receiving any plug-and-plug indicator from the operating system;

inspecting a driving program of the smart peripheral device using the plug-and-play notification and determining if the connected peripheral device is supported; and

if the connected peripheral device is supported,

using the plug-and-play notification to open the function device object established by a factory-provided driving program;

using the function device object to retrieve a physical device object established through the bus-driving program;

using the physical device object to retrieve a plurality of descriptors and calling data of the connected peripheral device; and

using the descriptors and calling data to set up necessary data for executing the driving program of the smart peripheral device.

2. The processing method of claim 1, wherein requesting that the operating system return any data regarding the connectivity status of a particular peripheral device connected to the peripheral bus further includes registering the request under a plug-and-play administrator within the operating system and requesting that a classification

identification for the connected peripheral device matching the peripheral bus be returned.

3. The processing method of claim 1, wherein receiving the plug-and-play notification from the operating system occurs when the peripheral device has already plugged into the peripheral bus or the peripheral device has just been plugged into the peripheral bus, and the operating system utilizes the plug-and-play notification to inform the driving program.

4. The processing method of claim 1, wherein using the plug-and-play notification to inspect whether the driving program supports the connected peripheral device or not includes using a special identification code in the plug-and-play notification to inspect the connected peripheral device and determine if the connected peripheral device matches the smart peripheral device to be processed by the driving program.

5. The processing method of claim 4, wherein:

if the peripheral bus is a Universal Serial Bus (USB), the special identification code is a vender identification and a product identification; and

if the peripheral bus is an IEEE-1394 bus, the special identification code is a plug-and-play identification.

6. The processing method of claim 1, wherein using the plug-and-play notification to open the function device object established through the factory-provided driving program includes using a device name within the plug-and-play notification corresponding to the connected peripheral device.

7. The processing method of claim 1, wherein using the function device object to retrieve the physical device object established through the bus-driving program further includes:

preparing a special input/output request packet and calling the function device object; and

retrieving the physical device object according to a pointer returned from the function device object inside the special input/output request packet.

8. The processing method of claim 1, wherein if the peripheral bus is a Universal Serial Bus (USB), the descriptors and calling data include a device descriptor, a configuration descriptor, an interface descriptor, an end-point descriptor and a USB pipe handles.

9. The processing method of claim 1, wherein if the peripheral bus is an IEEE-1394 bus, the descriptor and the calling data are stored a configuration read-only-memory.

10. The processing method of claim 1, wherein if the received plug-and-play notification indicates an absent peripheral device, processing includes:

inspecting whether the peripheral device is really absent;

informing through calling an application program of the driving program of the peripheral device that the peripheral device is absent;

canceling any in-process request on the absent peripheral device; and

returning the operating system to an initial state.

11. The processing method of claim 10, wherein the non-existence of the peripheral device means either a shutdown of power to the peripheral device or a removal of the peripheral device from the peripheral bus.

12. The processing method of claim 1, wherein the peripheral bus is either a Universal Serial Bus (USB) or an IEEE-1394 bus.

13. The processing method of claim 1, wherein if an application program used by the computer system issues a processing request to call up the driving program of the smart peripheral device, the processing method further includes:

inspecting the connected peripheral device to check if the peripheral device really exists,

if the peripheral device is absent, responding by issuing an error signal to the application program, and

if the peripheral device is not absent, determining if any processing request corresponding thereto and any end point for the connected peripheral device;

if the peripheral device does not have any end point, responding by returning the error signal to the application program; and

if the peripheral device has end point, then:

setting up a request block;

setting up an input/output request packet;

transmitting the input/output request packet to the physical device object for further processing,

when the physical device object finishes processing, responding by sending a correct signal to the application program; and

when the physical device object encounters a delay, responding by sending a delay signal to the application program.

14. A computer system that can be connected to a smart peripheral device, the computer system comprising:

a peripheral bus for connecting the smart peripheral device to the computer system;

an operating system having a bus-driving program for controlling the peripheral bus, wherein the bus-driving bus includes a physical device object for corresponding with the smart peripheral device;

a factory-provided driving program having a function device object, wherein the function device object communicates with the smart peripheral device through the physical device object; and

a general-purpose driving program for communicating with the smart peripheral device through the physical device object, wherein if the smart peripheral device is connected to the computer system, the general-purpose driving program switches on the function device object established through the factory-provided driving program, retrieves the physical device object established through the bus-driving program according to the function device object, retrieves a plurality of descriptors and calling data of the smart peripheral device according to the physical device object and finally sets up necessary information for executing the general-purpose driving program.

15. The computer system of claim 14, wherein the peripheral bus is either a Universal Serial Bus (USB) or an IEEE-1394 bus.

16. The computer system of claim 14, wherein the system further includes an application program such that when the application program issues a processing request call to the general-purpose driving program, the general-purpose driving program sets up a request block and an input/output request packet, the input/output request packet is sent to the physical device object for further processing, and after processing, the physical device object responds by sending a correct signal to the application program, and if the processing encounters a delay, the physical device object responds by sending a delay signal to the application program.